



# **Cisco Virtualized Infrastructure Manager Baremetal Manager API Guide, Release 3.4.1**

**First Published:** October 30, 2019

**Cisco Systems, Inc.**  
[www.cisco.com](http://www.cisco.com)

Cisco has more than 200 offices worldwide.  
Addresses, phone numbers, and fax numbers  
are listed on the Cisco website at  
[www.cisco.com/go/offices](http://www.cisco.com/go/offices).

# Cisco VIM Baremetal Manager API Guide

---

## Abstract

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: [www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks). Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Cisco Virtualization Infrastructure Manager Baremetal Manager API Guide  
© 2019 Cisco Systems, Inc. All rights reserved.

# Cisco VIM Baremetal Manager API Guide

## Contents

1	Overview .....	3
2	Usecases of REST API .....	3
3	Handling Security .....	4
4	Accessing Cisco VIM Baremetal Manager API .....	4
5	API Resources .....	4
6	Cisco VIM Baremetal Manager API Reference.....	4

## 1 Overview

The Cisco VIM Baremetal Manager provides a Representational State Transfer (REST) API that is used to deploy, expand, and manage RIMN.

An API conforms to the RESTful conventions and is defined by using resource and methods. A resource is a collection of information that is identified by a Uniform Resource Identifier (URI). Methods are HTTP methods that are exposed for a resource. The commonly used HTTP methods are POST, GET, PUT, and DELETE.

## 2 Usecases of REST API

The REST API provides a logical grouping of management nodes in form of site, cluster and nodes for better management of nodes globally.

```
site
  |-- clusters
      |-- cluster_0
          |-- servers
              |-- node_0.0
              |.
              |.
              |-- node_0.n
          |.
          |.
          |-- cluster_n
              |-- servers
                  |-- node_n.0
                  |.
                  |.
                  |-- node_n.n
```

The following list summarizes the various actions that can be performed using REST API:

- Import ISO files for booting Management node.
- Deploy a Cisco VIM Baremetal Manager site, cluster, and node.
- Add cluster to a deployed site.
- Delete cluster from a deployed site.
- Add node to a deployed cluster.
- Delete node from a deployed cluster.
- Jobs to track deployment of site, cluster, and node.

# Cisco VIM Baremetal Manager API Guide

---

## 3 Handling Security

The Cisco VIM Baremetal Manager REST API security is provided by the Secure Sockets Layer (SSL) included on the Apache Web Server. The Flask-Restplus based web application runs the Rest API server.

The Flask REST API server requires a username and password to authorize the REST API server requests. Apache handles the authorization process, which authorizes the request to access the Flask web application.

## 4 Accessing Cisco VIM Baremetal Manager API

You can access API server on the br\_api interface on port 8141. Authentication is enabled by default in the webservice.

You can access the API end points of a version (v1 now) using the following URL format: [https://<management\\_node\\_api\\_ip>:8141/v1](https://<management_node_api_ip>:8141/v1)

Where, <management\_node\_api\_ip> is the virtual IP address that you provided during the installation of Cisco VIM management node.

By default, the basic authentication is enabled for the API endpoints in the management node.

In the management node, you can find the authentication credentials in the client\_config.json file available at the location:

```
opt/cisco/argus/rest_api/client_config.json
```

The following code shows a sample content of the client\_config.json file:

```
{
  "REST_API_URL": "http://10.22.191.134:8141",
  "REST_API_USERNAME": "admin",
  "REST_API_PASSWORD": "8675d63674ff686e8688",
  "PODTYPE": "rmi"
}
```

## 5 API Resources

Cisco VIM Baremetal manager REST API is a Flask-Restplus based web application which comes with Swagger integration. Swagger is built around OpenAPI specification that helps to design, build, document, and consume REST APIs.

The REST API resources along with their expected payloads and responses have been documented by Swagger.

## 6 Cisco VIM Baremetal Manager API Reference

Here is the view of Cisco VIM Baremetal Manager REST API:

# Argus<sup>v1</sup>

[ Base URL: /v1 ]

Argus: Cisco's Baremetal Installer

## site Argus site operations

**PUT** `/site/{site_name}` Alter a site given its name and action

Try it out

Name	Description
<b>action</b> * required string <i>(query)</i>	The argus site action identifier
<b>site_name</b> * required string <i>(path)</i>	The argus site identifier

Response content type application/json

Code	Description
202	<i>Site requested operation accepted</i>  Example Value <b>Model</b>  <pre> Job Action Response {   status*      string   resource_name* string   job_uuid*    string }                     </pre>
404	<i>Site not found</i>
409	<i>Conflict in performing request</i>

**POST** `/site/{site_name}` Create a site given its name and payload

Try it out

Name	Description
<b>payload</b> * required (body)	Example Value <b>Model</b>  <pre> Site {   info* }                     </pre>

Name

Description

```
string
minLength: 1
maxLength: 50
Site info
clusters*
  [Cluster info* {
    string
    minLength: 1
    maxLength: 50
    Cluster info
    name*
    string
    minLength: 1
    maxLength: 32
    Cluster name
    servers*
    [Node {
      name*
      string
      minLength: 1
      maxLength: 32
      Node name
      oob_ip*
      string
      OOB/CIMC IP
      oob_username*
      string
      OOB/CIMC username
      domain_name_servers
      [string]
      oob_password*
      string
      OOB/CIMC password
      flavor*
      string
      Flavor name
      boot_network
      string
      Boot Network interface
      ip_address*
      Node Interface {
        api_[1-9]+_v6
        string
        API IPv6 CIDR
        api_[1-9]
        +_v4*
        string
        API IPv4 CIDR
        management_
        [1-9]+_v6
        string
        Management IPv6 CIDR
        management_
        [1-9]+_v4*
        string
        Management IPv4 CIDR
      }
      password_hash
      string
      Password Hash
    }
  }
}
networking*
  Networking {
    ntp_servers*
    [string]
    description: NTP IP
    domain_name_servers*
    [string]
    description: DNS IP
    domain_name*
    string
    Domain Name
    networks*
    Networks {
      api_[1-9]+_v6
      Network Interface {
        subnet*
        string
        Subnet CIDR
        gateway
        string
        Gateway IP
        vlan_id
        integer
        VLAN ID
      }
      api_[1-9]
      +_v4*
      Network Interface {
        subnet*
        string
        Subnet CIDR
        gateway
        string
        Gateway IP
        vlan_id
        integer
        VLAN ID
      }
      management_
      [1-9]+_v6
      Network Interface {
        subnet*
        string
        Subnet CIDR
        gateway
        string
        Gateway IP
        vlan_id
        integer
        VLAN ID
      }
      management_
      [1-9]+_v4*
      Network Interface {
        subnet*
        string
        Subnet CIDR
        gateway
        string
        Gateway IP
        vlan_id
        integer
        VLAN ID
      }
    }
  }
}
```

Name	Description
name*	<pre>string minLength: 1 maxLength: 32</pre>
metadata*	<p><b>Site name</b></p> <pre>Metadata {   ssh_key* string   password_hash* string }</pre> <p><b>Password Hash</b></p>
site_name* <small>required</small>	The argus site identifier
string (path)	

**Responses** Response content type

Code	Description
201	Site created successfully
400	Bad request
409	Conflict in performing request

**GET** /site/{site\_name} Fetch a site given its name

[Try it out](#)

Name	Description
site_name* <small>required</small>	The argus site identifier
string (path)	

**Responses** Response content type

Code	Description
200	Success

Example Value **Model**

```
Site Response {
  info* string
    minLength: 1
    maxLength: 50
    Site info

  status* string
  networking* Networking Response {
    ntp_servers* [string]
      description: NTP IP
    domain_name_servers*
```

Code Description

```
[string]
description: DNS IP
domain_name* string
Domain Name

networks* Networks Response {
  api_[1-9]+_v6 Network configs Response {
    subnet* string
    Subnet CIDR

    gateway string
    Gateway IP

    vlan_id integer
    VLAN ID
  }

  api_[1-9]+_v4* Network configs Response {
    subnet* string
    Subnet CIDR

    gateway string
    Gateway IP

    vlan_id integer
    VLAN ID
  }

  management_[1-9]+_v6 Network configs Response {
    subnet* string
    Subnet CIDR

    gateway string
    Gateway IP

    vlan_id integer
    VLAN ID
  }

  management_[1-9]+_v4* Network configs Response {
    subnet* string
    Subnet CIDR

    gateway string
    Gateway IP

    vlan_id integer
    VLAN ID
  }
}

name* string
minLength: 1
maxLength: 32
Site name

clusters* Clusters Response {
  <cluster_name>* Cluster Config Response {
    info* string
    minLength: 1
    maxLength: 50
    Cluster info

    status* string

    servers* Nodes Response {
      <node_name>* Node Response {
        status* string

        oob_ip* string
        OOB/CIMC IP

        oob_username* string
        OOB/CIMC username

        domain_name_servers [string]

        oob_password* string
        OOB/CIMC password

        flavor* string
        Flavor name

        boot_network string
        Boot Network interface

        ip_address* Node Interface Response
          {
            api_[1-9]+_v6 string
            API IPv6 CIDR

            api_[1-9]+_v4* string
            API IPv4 CIDR

            string
          }
      }
    }
  }
}
```





No parameters

Responses

Response content type

application/json

Code

Description

200

Success

Example Value Model

```
Site List Response {
  <site_name>* Site Info Response {
    info* string
              minLength: 1
              maxLength: 50
              Site info
    status* string
  }
}
```

## cluster Argus cluster operations

GET /cluster/{site\_name} List all clusters in the site

Parameters

Try it out

Name

Description

site\_name \* required

The argus site identifier

string  
(path)

Responses

Response content type

application/json

Code

Description

200

Success

Example Value Model

```
Cluster List Response {
  <cluster_name>* Cluster Info Response {
    info* string
           minLength: 1
           maxLength: 50
           Cluster info
    status* string
  }
}
```

PUT /cluster/{site\_name}/{cluster\_name} Alter a cluster given its name and action

Parameters

Try it out

Name	Description
<b>action</b> * required string (query)	The argus cluster action identifier
<b>cluster_name</b> * required string (path)	The argus cluster identifier
<b>site_name</b> * required string (path)	The argus site identifier

  

Responses	Response content type
	application/json

  

Code	Description
202	<i>Cluster requested operation accepted</i>
404	<i>Cluster not found</i>
409	<i>Conflict in performing request</i>

**POST** /cluster/{site\_name}/{cluster\_name} Create a cluster given its name, site and payload

**Parameters** Try it out

Name	Description
<b>payload</b> * required (body)	<div style="display: flex; align-items: flex-start;"> <div style="margin-right: 20px;"> <p>Example Value</p> <pre>Site {   info*   clusters* }</pre> </div> <div> <p><b>Model</b></p> <pre>string minLength: 1 maxLength: 50 Site info [Cluster {   info*   name*   servers* } string minLength: 1 maxLength: 50 Cluster info string minLength: 1 maxLength: 32 Cluster name [Node {   name*   oob_ip*   oob_username* } string minLength: 1 maxLength: 32 Node name string OOB/CIMC IP string OOB/CIMC username</pre> </div> </div>

Name Description

```
domain_name_servers [string]
oob_password* string OOB/CIMC password
flavor* string Flavor name
boot_network string Boot Network interface
ip_address* Node Interface {
  api_[1-9]+_v6 string API IPv6 CIDR
  api_[1-9]+_v4* string API IPv4 CIDR
  management_[1-9]+_v6 string Management IPv6 CIDR
  management_[1-9]+_v4* string Management IPv4 CIDR
}
password_hash string Password Hash
}}
networking* Networking {
  ntp_servers* [string] description: NTP IP
  domain_name_servers* [string] description: DNS IP
  domain_name* string Domain Name
  networks* Networks {
    api_[1-9]+_v6 Network Interface {
      subnet* string Subnet CIDR
      gateway string Gateway IP
      vlan_id integer VLAN ID
    }
    api_[1-9]+_v4* Network Interface {
      subnet* string Subnet CIDR
      gateway string Gateway IP
      vlan_id integer VLAN ID
    }
    management_[1-9]+_v6 Network Interface {
      subnet* string Subnet CIDR
      gateway string Gateway IP
      vlan_id integer VLAN ID
    }
    management_[1-9]+_v4* Network Interface {
      subnet* string Subnet CIDR
      gateway string Gateway IP
      vlan_id integer VLAN ID
    }
  }
}
name* string minLength: 1 maxLength: 32 Site name
metadata* Metadata {
  ssh_key* string
  password_hash* string Password Hash
}
}
```

**cluster\_name** \* required The argus cluster identifier  
string  
(path)

**site\_name** \* required The argus site identifier  
string  
(path)

## Responses

Response content type

application/json

### Code

### Description

201	<i>Cluster created successfully</i>
400	<i>Bad request</i>
403	<i>Request not allowed</i>
409	<i>Conflict in performing request</i>

GET /cluster/{site\_name}/{cluster\_name} Fetch a cluster given its name and site

## Parameters

Try it out

### Name

### Description

<b>cluster_name</b> * required string (path)	The argus cluster identifier
<b>site_name</b> * required string (path)	The argus site identifier

## Responses

Response content type

application/json

### Code

### Description

200	<i>Success</i>
-----	----------------

Example Value Model

```
Clusters Response {
  <cluster_name>* Cluster Config Response {
    info* string
      minLength: 1
      maxLength: 50
      Cluster info
    status* string
    servers* Nodes Response {
      <node_name>* Node Response {
        status* string
        oob_ip* string
          OOB/CIMC IP
        oob_username* string
          OOB/CIMC username
        domain_name_servers [string]
        oob_password* string
          OOB/CIMC password
        flavor* string
          Flavor name
      }
    }
  }
}
```

Code	Description
	<pre> boot_network string Boot Network interface  ip_address* Node Interface Response {   api_[1-9]+_v6 string                     API IPv6 CIDR    api_[1-9] +_v4* string         API IPv4 CIDR    management_ [1-9]+_v6 string         Management IPv6 CIDR    management_ [1-9]+_v4* string         Management IPv4 CIDR }  password_hash string Password Hash } } } } } </pre>
404	<i>Cluster not found</i>

DELETE /cluster/{site_name}/{cluster_name} Delete a cluster given its name and site	
<div style="float: right; border: 1px solid black; padding: 2px 5px;">Try it out</div>	
<b>Parameters</b>	
Name	Description
<b>cluster_name</b> * required string (path)	The argus cluster identifier
<b>site_name</b> * required string (path)	The argus site identifier
<b>Responses</b> <div style="float: right; border: 1px solid black; padding: 2px 5px;">application/json</div>	
Code	Description
202	<i>Cluster deleted operation accepted</i>
	<div style="display: flex; justify-content: space-between;"> <span>Example Value</span> <span>Model</span> </div> <pre> Job Action Response {   status* string   resource_name* string   job_uuid* string } </pre>
204	<i>Cluster deleted successfully</i>
404	<i>Cluster not found</i>
409	<i>Conflict in performing request</i>

## node Argus node operations

**PUT** `/node/{site_name}/{cluster_name}/{node_name}` Alter a node given its name, cluster, site and action

**Parameters** Try it out

Name	Description
<b>action</b> * required string (query)	The argus node action identifier
<b>cluster_name</b> * required string (path)	The argus cluster identifier
<b>site_name</b> * required string (path)	The argus site identifier
<b>node_name</b> * required string (path)	The argus node identifier

**Responses** Response content type: application/json

Code	Description
202	<i>Node requested operation accepted</i>
404	<i>Node not found</i>
409	<i>Conflict in performing request</i>

Example Value **Model**

```
Job Action Response {
  status*      string
  resource_name* string
  job_uuid*    string
}
```

**POST** `/node/{site_name}/{cluster_name}/{node_name}` Create a node given its name, cluster, site and payload

**Parameters** Try it out

Name	Description
<b>payload</b> * required string (body)	Example Value <b>Model</b>

```
Site {
  info*      string
             minLength: 1
             maxLength: 50
  Site info
  clusters*  [Cluster {
             info*
```

Name	Description
	<pre> string minLength: 1 maxLength: 50 Cluster info name* string minLength: 1 maxLength: 32 Cluster name servers* [Node { name* string minLength: 1 maxLength: 32 Node name oob_ip* string OOB/CIMC IP oob_username* string OOB/CIMC username domain_name_servers [string] oob_password* string OOB/CIMC password flavor* string Flavor name boot_network string Boot Network interface ip_address* Node Interface { api_[1-9]+_v6 string API IPv6 CIDR api_[1-9] +_v4* string API IPv4 CIDR management_ [1-9]+_v6 string Management IPv6 CIDR management_ [1-9]+_v4* string Management IPv4 CIDR } password_hash string Password Hash } }] networking* Networking { ntp_servers* [string] description: NTP IP domain_name_servers* [string] description: DNS IP domain_name* string Domain Name networks* Networks { api_[1-9]+_v6 Network Interface { subnet* string Subnet CIDR gateway string Gateway IP vlan_id integer VLAN ID } api_[1-9] +_v4* Network Interface { subnet* string Subnet CIDR gateway string Gateway IP vlan_id integer VLAN ID } management_ [1-9]+_v6 Network Interface { subnet* string Subnet CIDR gateway string Gateway IP vlan_id integer VLAN ID } management_ [1-9]+_v4* Network Interface { subnet* string Subnet CIDR gateway string Gateway IP vlan_id integer VLAN ID } } } name* string minLength: 1 maxLength: 32 Site name metadata* </pre>



Name	Description
	<pre> Metadata {   ssh_key* string   password_hash* string   Password Hash } </pre>
<b>cluster_name</b> * required	The argus cluster identifier
string (path)	
<b>site_name</b> * required	The argus site identifier
string (path)	
<b>node_name</b> * required	The argus node identifier
string (path)	
<b>Responses</b>	Response content type <input type="text" value="application/json"/>
Code	Description
201	<i>Node created successfully</i>
400	<i>Bad request</i>
403	<i>Request not allowed</i>
409	<i>Conflict in performing request</i>

GET /node/{site_name}/{cluster_name}/{node_name} Fetch a node given its name, cluster and site	
<b>Parameters</b>	<input type="button" value="Try it out"/>
Name	Description
<b>cluster_name</b> * required	The argus cluster identifier
string (path)	
<b>site_name</b> * required	The argus site identifier
string (path)	
<b>node_name</b> * required	The argus node identifier
string (path)	
<b>Responses</b>	Response content type <input type="text" value="application/json"/>
Code	Description

Code	Description
200	<i>Success</i>
	<p>Example Value <b>Model</b></p> <pre> Nodes Response {   &lt;node_name&gt;* Node Response {     status*      string     oob_ip*      string                   OOB/CIMC IP      oob_username* string                   OOB/CIMC username      domain_name_servers [string]     oob_password*      string                   OOB/CIMC password      flavor*      string                   Flavor name      boot_network string                   Boot Network interface      ip_address*  Node Interface Response {       api_[1-9]+_v6 string                   API IPv6 CIDR        api_[1-9] +_v4*          string                   API IPv4 CIDR        management_ [1-9]+_v6     string                   Management IPv6 CIDR        management_ [1-9]+_v4*    string                   Management IPv4 CIDR     }      password_hash string                   Password Hash   } } </pre>
404	<i>Resource not found</i>

**DELETE** `/node/{site_name}/{cluster_name}/{node_name}` Delete a node given its name, cluster and site

**Parameters** Try it out

Name	Description
<b>cluster_name</b> * required string (path)	The argus cluster identifier
<b>site_name</b> * required string (path)	The argus site identifier
<b>node_name</b> * required string (path)	The argus node identifier

**Responses** Response content type application/json

Code	Description
202	<i>Node delete operation accepted</i>

Code	Description
	<p>Example Value    <b>Model</b></p> <pre> Job Action Response  {   status*             string   resource_name*      string   job_uuid*           string } </pre>
204	<i>Node deleted successfully</i>
404	<i>Node not found</i>
409	<i>Conflict in performing request</i>

**GET**    `/node/{site_name}/{cluster_name}`    List all nodes in the cluster of a site

**Parameters** Try it out

Name	Description
<b>cluster_name</b> * required string <i>(path)</i>	The argus cluster identifier
<b>site_name</b> * required string <i>(path)</i>	The argus site identifier

**Responses** Response content type application/json

Code	Description
200	<i>Success</i>
	<p>Example Value    <b>Model</b></p> <pre> Node List Response  {   &lt;node_name&gt;*      Node Info Response  {                         status*             string                       } } </pre>

## job Argus job operations

**GET**    `/job/jobs_info`    Fetch all ToRun/Running/Aborting jobs

**Parameters** Try it out

No parameters

**Responses** Response content type

application/json

Code	Description
200	<i>Success</i>

Example Value    **Model**

```
Running Jobs Response {
  <job_uuid>* Job Show Response {
    tasks* [Task Response {
      status* string
      updated_at* string($date-time)
      server* string
      command* string
      error* string
      started_at* string($date-time)
      stage* string
    }]
    description* string
    created_at* string($date-time)
    updated_at* string($date-time)
    aborted_at* string($date-time)
    version* string
    command* string
    log* string
    error* string
    started_at* string($date-time)
    stage* string
  }
}
```

**PUT**    `/job/{uuid}` Alter a job given its uuid and action

**Parameters**

Try it out

Name	Description
<b>action</b> * required string <i>(query)</i>	The argus job action identifier
<b>uuid</b> * required string <i>(path)</i>	The argus job identifier

**Responses**

Response content type    application/json

Code	Description
202	<i>Job requested operation accepted</i>
404	<i>Job not found</i>
409	<i>Conflict in performing request</i>

**GET**    `/job/{uuid}` Fetch a job given its uuid

**Parameters**

Try it out

Name	Description
<b>uuid</b> * required string (path)	The argus job identifier

Responses Response content type

Code	Description
200	<p><i>Success</i></p> <p>Example Value    <b>Model</b></p> <pre>Job Show Response {   tasks* [Task Response {     status* string     updated_at* string(\$date-time)     server* string     command* string     error* string     started_at* string(\$date-time)     stage* string   }]   description* string   created_at* string(\$date-time)   updated_at* string(\$date-time)   aborted_at* string(\$date-time)   version* string   command* string   log* string   error* string   started_at* string(\$date-time)   stage* string }</pre>
404	<p><i>Job not found</i></p>

DELETE /job/{uuid} Delete a job forcefully

Try it out

Name	Description
<b>uuid</b> * required string (path)	The argus job identifier

Responses Response content type

Code	Description
204	<p><i>Job deleted successfully</i></p>
404	<p><i>Job not found</i></p>

Code	Description
------	-------------

**GET** `/job/` List all Jobs [Try it out](#)

**Parameters**

No parameters

**Responses** Response content type `application/json`

Code	Description
200	<i>Success</i>

Example Value **Model**

```

Job List Response {
  <job_uuid>* Job Info Response {
    status* string
    stage* string
    description* string
    updated_at* string($date-time)
  }
}

```

## flavor Argus flavor operations

**GET** `/flavor/{flavor_name}` Fetch a flavor given its name [Try it out](#)

**Parameters**

Name	Description
<b>flavor_name</b> * required string (path)	The argus flavor identifier

**Responses** Response content type `application/json`

Code	Description
200	<i>Success</i>

Example Value **Model**

```

Flavor Show Response {
  name* string
  workflow* string
  boot* Boot Response {
    mode* string
  }
  os_policies* [string]
  disk_policies* [string]
  network_policies*
}

```

Code	Description
404	<pre> } Flavor not found </pre>

**GET** `/flavor/` List all flavors

**Parameters** Try it out

No parameters

**Responses** Response content type: application/json

Code	Description
200	<p>Success</p> <p>Example Value <b>Model</b></p> <pre> Flavor List Response {   flavors* [string] } </pre>

**iso** Argus ISO operations

**POST** `/iso/{iso_name}` Import a ISO given its name

**Parameters** Try it out

Name	Description
<p><b>payload</b> * required</p> <p>(body)</p>	<p>Example Value <b>Model</b></p> <pre> ISO Validation {   file_name* string   ISO File Name } </pre>
<p><b>iso_name</b> * required</p> <p>string</p> <p>(path)</p>	<p>The argus iso identifier</p>

**Responses** Response content type: application/json

Code	Description
201	<p>ISO imported successfully</p>

Code	Description
	<p>Example Value    <b>Model</b></p> <pre> ISO Create Response {   flavor*      string } </pre>
400	<i>Bad request</i>
409	<i>Conflict in performing request</i>

**GET**    `/iso/{iso_name}`    Fetch a ISO given its name

**Parameters** Try it out

Name	Description
<b>iso_name</b> * required string (path)	The argus iso identifier

**Responses** Response content type

Code	Description
200	<i>Success</i>
	<p>Example Value    <b>Model</b></p> <pre> ISO Show Response {   file_name*  string   flavor*    string } </pre>
404	<i>ISO not found</i>

**DELETE**    `/iso/{iso_name}`    Delete a ISO forcefully

**Parameters** Try it out

Name	Description
<b>iso_name</b> * required string (path)	The argus iso identifier

**Responses** Response content type

Code	Description
------	-------------



Code	Description
204	<i>ISO deleted successfully</i>
404	<i>ISO not found</i>

GET `/iso/` List all ISOs

Parameters Try it out

No parameters

Responses Response content type `application/json`

Code	Description
200	<i>Success</i>

Example Value **Model**

```

ISO List Response {
  isos* [string]
}

```

## setupdata Baremetal Site to Site Translator

POST `/setupdata/{site_name}` Translate a site config to Argus standards

Parameters Try it out

Name	Description
<b>payload</b> * required <i>(body)</i>	<p>Example Value <b>Model</b></p> <pre> Baremetal Site Payload {   info* string     minLength: 1     maxLength: 50   Site info   clusters* [Baremetal Cluster Payload] {     info* string       minLength: 1       maxLength: 50     Cluster info     name* string       minLength: 1       maxLength: 32     Cluster name     servers* [Baremetal Node Payload] {       name* string         minLength: 1         maxLength: 32       Node name       oob_ip* string         OOB/CIMC IP       oob_username string         OOB/CIMC username       domain_name_servers [string]       oob_password string         OOB/CIMC password     }   } } </pre>

Name	Description
flavor	string <b>Flavor name</b>
boot_network	string <b>Boot Network interface</b>
ip_address*	Baremetal Node Interface Payload { api_[1-9] integer +_vlan_id <b>API VLAN ID</b> api_[1-9] string +_gateway_v4* <b>API IPv4 Gateway</b> api_[1-9]+_v6 string <b>API IPv6 CIDR</b> api_[1-9] string +_v4* <b>API IPv4 CIDR</b> api_[1-9] string +_gateway_v6 <b>API IPv6 Gateway</b> management_ [1-9]+_v6 <b>Management IPv6</b> management_ [1-9] string <b>Management IPv4 Gateway</b> +_gateway_v4 string management_ [1-9]+_v4* <b>Management IPv4</b> management_ [1-9] string <b>Management IPv6 Gateway</b> +_gateway_v6 }
password_hash	string <b>Password Hash</b>
name*	string minLength: 1 maxLength: 32 <b>Site name</b>
common_info*	Baremetal Common Info Payload { ntp_servers [string] description: NTP IP(s) oob_username* string <b>OOB/CIMC username</b> domain_name_servers* [string] description: DNS IP(s) ssh_key* string domain_name* string <b>Domain Name</b> oob_password* string <b>OOB/CIMC password</b> flavor* string <b>Flavor name</b> password_hash* string <b>Password Hash</b> }
cluster_name	The argus cluster identifier string (query)
site_name * required	The argus site identifier string (path)
node_name	The argus node identifier string (query)

**Responses** Response content type application/json

Code	Description
201	Translated successfully
Example Value	Model
Site	{

## Code

## Description

```

info*      string
           minLength: 1
           maxLength: 50
           Site info

clusters*  [Cluster {
           info*      string
                   minLength: 1
                   maxLength: 50
                   Cluster info

           name*      string
                   minLength: 1
                   maxLength: 32
                   Cluster name

           servers*   [Node {
                   name*      string
                           minLength: 1
                           maxLength: 32
                           Node name

                   oob_ip*    string
                           OOB/CIMC IP

                   oob_username* string
                           OOB/CIMC username

                   domain_name_servers [string]
                   oob_password*      string
                           OOB/CIMC password

                   flavor*      string
                           Flavor name

                   boot_network string
                           Boot Network interface

                   ip_address*   Node Interface {
                           api_[1-9]+_v6 string
                                   API IPv6 CIDR

                           api_[1-9]
                           +_v4*      string
                                   API IPv4 CIDR

                           management_
                           [1-9]+_v6  string
                                   Management IPv6 CIDR

                           management_
                           [1-9]+_v4* string
                                   Management IPv4 CIDR

                           }

                   password_hash string
                           Password Hash

                   }
           ]
}}

networking* Networking {
           ntp_servers* [string]
                   description: NTP IP

           domain_name_servers* [string]
                   description: DNS IP

           domain_name* string
                   Domain Name

           networks* Networks {
                   api_[1-9]+_v6 Network Interface {
                           subnet* string
                                   Subnet CIDR

                           gateway string
                                   Gateway IP

                           vlan_id integer
                                   VLAN ID

                           }

                   api_[1-9]
                   +_v4*      Network Interface {
                           subnet* string
                                   Subnet CIDR

                           gateway string
                                   Gateway IP

                           vlan_id integer
                                   VLAN ID

                           }

                   management_
                   [1-9]+_v6  Network Interface {
                           subnet* string
                                   Subnet CIDR

                           gateway string
                                   Gateway IP

                           vlan_id integer
                                   VLAN ID

                           }
           }
}

```

**Code**      **Description**

```

management_
[1-9]+_v4*
}
Network Interface {
  subnet* string
  Subnet CIDR

  gateway string
  Gateway IP

  vlan_id integer
  VLAN ID
}
}
}
name* string
minLength: 1
maxLength: 32
Site name
metadata* Metadata {
  ssh_key* string
  password_hash* string
  Password Hash
}
}
}

400 Bad request

409 Conflict in performing request

```

**Models**

```

Boot Response {
  mode* string
}

Cluster Config Response {
  info* string
  minLength: 1
  maxLength: 50
  Cluster info

  status* string
  servers* Nodes Response {
    <node_name>* Node Response {
      status* string
      oob_ip* string
      OOB/CIMC IP

      oob_username* string
      OOB/CIMC username

      domain_name_servers [string]
      oob_password* string
      OOB/CIMC password

      flavor* string
      Flavor name

      boot_network string
      Boot Network interface

      ip_address* Node Interface Response {
        api_[1-9]+_v6 string
        API IPv6 CIDR

        api_[1-9]
+_v4* string
API IPv4 CIDR

        management_
[1-9]+_v6 string
Management IPv6 CIDR

        management_
[1-9]+_v4* string
Management IPv4 CIDR
      }

      password_hash string
      Password Hash
    }
  }
}
}

```

```

    }
}

Networking {
  ntp_servers* [string]
  description: NTP IP

  domain_name_servers* [string]
  description: DNS IP

  domain_name* string
  Domain Name

  networks* Networks {
    api_[1-9]+_v6 Network Interface {
      subnet* string
      Subnet CIDR

      gateway string
      Gateway IP

      vlan_id integer
      VLAN ID
    }

    api_[1-9]+_v4* Network Interface {
      subnet* string
      Subnet CIDR

      gateway string
      Gateway IP

      vlan_id integer
      VLAN ID
    }

    management_[1-9]+_v6 Network Interface {
      subnet* string
      Subnet CIDR

      gateway string
      Gateway IP

      vlan_id integer
      VLAN ID
    }

    management_[1-9]+_v4* Network Interface {
      subnet* string
      Subnet CIDR

      gateway string
      Gateway IP

      vlan_id integer
      VLAN ID
    }
  }
}

```

```

ISO List Response {
  isos* [string]
}

```

```

Node Response {
  status* string

  oob_ip* string
  OOB/CIMC IP

  oob_username* string
  OOB/CIMC username

  domain_name_servers [string]

  oob_password* string
  OOB/CIMC password

  flavor* string
  Flavor name

  boot_network string
  Boot Network interface

  ip_address* Node Interface Response {
    api_[1-9]+_v6 string
    API IPv6 CIDR

    api_[1-9]+_v4* string
    API IPv4 CIDR

    management_[1-9]+_v6 string
    Management IPv6 CIDR

    management_[1-9]+_v4* string
    Management IPv4 CIDR
  }
}

```

```

    password_hash      }
                      string
                      Password Hash
}

Node Interface Response {
  api_[1-9]+_v6 string
                  API IPv6 CIDR

  api_[1-9]
+_v4* string
      API IPv4 CIDR

  management_
[1-9]+_v6 string
         Management IPv6 CIDR

  management_
[1-9]+_v4* string
         Management IPv4 CIDR
}

Node Info Response {
  status* string
}

Job List Response {
  <job_uuid>* Job Info Response {
    status* string
    stage* string
    description* string
    updated_at* string($date-time)
  }
}

Networking Response {
  ntp_servers* [string]
               description: NTP IP

  domain_name_servers* [string]
                       description: DNS IP

  domain_name* string
              Domain Name

  networks* Networks Response {
    api_[1-9]+_v6 Network configs Response {
      subnet* string
              Subnet CIDR

      gateway string
              Gateway IP

      vlan_id integer
              VLAN ID
    }

    api_[1-9]
+_v4* Network configs Response {
      subnet* string
              Subnet CIDR

      gateway string
              Gateway IP

      vlan_id integer
              VLAN ID
    }

    management_
[1-9]+_v6 Network configs Response {
      subnet* string
              Subnet CIDR

      gateway string
              Gateway IP

      vlan_id integer
              VLAN ID
    }

    management_
[1-9]+_v4* Network configs Response {
      subnet* string
              Subnet CIDR

      gateway string
              Gateway IP

      vlan_id integer
              VLAN ID
    }
  }
}

```

```

Site Info Response {
  info*      string
             minLength: 1
             maxLength: 50
             Site info

  status*    string
}

```

```

Metadata {
  ssh_key*   string
  password_hash* string
             Password Hash
}

```

```

Node {
  name*      string
             minLength: 1
             maxLength: 32
             Node name

  oob_ip*    string
             OOB/CIMC IP

  oob_username* string
             OOB/CIMC username

  domain_name_servers [string]
  oob_password*      string
             OOB/CIMC password

  flavor*    string
             Flavor name

  boot_network string
             Boot Network interface

  ip_address* Node Interface {
                 api_[1-9]+_v6 string
                 API IPv6 CIDR

                 api_[1-9]
                 +_v4* string
                 API IPv4 CIDR

                 management_
                 [1-9]+_v6 string
                 Management IPv6 CIDR

                 management_
                 [1-9]+_v4* string
                 Management IPv4 CIDR
             }

  password_hash string
             Password Hash
}

```

```

Cluster Info Response {
  info*      string
             minLength: 1
             maxLength: 50
             Cluster info

  status*    string
}

```

```

Baremetal Common Info Payload {
  ntp_servers [string]
             description: NTP IP(s)

  oob_username* string
             OOB/CIMC username

  domain_name_servers* [string]
             description: DNS IP(s)

  ssh_key*      string
  domain_name* string
             Domain Name

  oob_password* string
             OOB/CIMC password

  flavor*       string
             Flavor name

  password_hash* string
             Password Hash
}

```

```

Network configs Response {

```

```

subnet*      string
             Subnet CIDR

gateway      string
             Gateway IP

vlan_id     integer
            VLAN ID

}

Clusters Response {
  <cluster_name>* Cluster Config Response {
    info*      string
               minLength: 1
               maxLength: 50
               Cluster info

    status*   string

    servers*  Nodes Response {
      <node_name>* Node Response {
        status*      string
        oob_ip*      string
                     OOB/CIMC IP

        oob_username* string
                     OOB/CIMC username

        domain_name_servers [string]
        oob_password*      string
                     OOB/CIMC password

        flavor*      string
                     Flavor name

        boot_network string
                     Boot Network interface

        ip_address*  Node Interface Response {
          api_[1-9]+_v6 string
                       API IPv6 CIDR

          api_[1-9]
          +_v4*      string
                     API IPv4 CIDR

          management_ string
          [1-9]+_v6   Management IPv6 CIDR

          management_ string
          [1-9]+_v4*  Management IPv4 CIDR
        }

        password_hash string
                       Password Hash
      }
    }
  }
}

Task Response {
  status*      string
  updated_at*  string($date-time)
  server*      string
  command*     string
  error*       string
  started_at*  string($date-time)
  stage*       string
}

Job Action Response {
  status*      string
  resource_name* string
  job_uuid*    string
}

Job Info Response {
  status*      string
  stage*       string
  description* string
  updated_at*  string($date-time)
}

ISO Create Response {
  flavor*      string
}

```



```

Flavor List Response {
  flavors* [string]
}

ISO Validation {
  file_name* string
  ISO File Name
}

Baremetal Node Payload {
  name* string
  minLength: 1
  maxLength: 32
  Node name

  oob_ip* string
  OOB/CIMC IP

  oob_username string
  OOB/CIMC username

  domain_name_servers [string]
  oob_password string
  OOB/CIMC password

  flavor string
  Flavor name

  boot_network string
  Boot Network interface

  ip_address* Baremetal Node Interface Payload {
    api_[1-9] integer
    +_vlan_id API VLAN ID

    api_[1-9] string
    +_gateway_v4* API IPv4 Gateway

    api_[1-9]+_v6 string
    API IPv6 CIDR

    api_[1-9] string
    +_v4* API IPv4 CIDR

    api_[1-9] string
    +_gateway_v6 API IPv6 Gateway

    management_[1-9]+_v6 string
    Management IPv6

    management_[1-9] string
    +_gateway_v4 Management IPv4 Gateway

    management_[1-9]+_v4* string
    Management IPv4

    management_[1-9] string
    +_gateway_v6 Management IPv6 Gateway
  }

  password_hash string
  Password Hash
}

Flavor Show Response {
  name* string
  workflow* string
  boot* Boot Response {
    mode* string
  }

  os_policies* [string]
  disk_policies* [string]
  network_policies* [string]
}

Network Interface {
  subnet* string
  Subnet CIDR

  gateway string
  Gateway IP

  vlan_id integer
  VLAN ID
}

```

```

Metadata Response {
  ssh_key* string
  password_hash* string
  Password Hash
}

```

```

Baremetal Node Interface Payload {
  api_[1-9] integer
  +_vlan_id API VLAN ID

  api_[1-9] string
  +_gateway_v4* API IPv4 Gateway

  api_[1-9]+_v6 string
  API IPv6 CIDR

  api_[1-9] string
  +_v4* API IPv4 CIDR

  api_[1-9] string
  +_gateway_v6 API IPv6 Gateway

  management_ string
  [1-9]+_v6 Management IPv6

  management_ string
  [1-9] Management IPv4 Gateway
  +_gateway_v4

  management_ string
  [1-9]+_v4* Management IPv4

  management_ string
  [1-9] Management IPv6 Gateway
  +_gateway_v6
}

```

```

Nodes Response {
  <node_name>* Node Response {
    status* string
    oob_ip* string
    OOB/CIMC IP

    oob_username* string
    OOB/CIMC username

    domain_name_servers [string]
    oob_password* string
    OOB/CIMC password

    flavor* string
    Flavor name

    boot_network string
    Boot Network interface

    ip_address* Node Interface Response {
      api_[1-9]+_v6 string
      API IPv6 CIDR

      api_[1-9] string
      +_v4* API IPv4 CIDR

      management_ string
      [1-9]+_v6 Management IPv6 CIDR

      management_ string
      [1-9]+_v4* Management IPv4 CIDR
    }

    password_hash string
    Password Hash
  }
}

```

```

ISO Show Response {
  file_name* string
  flavor* string
}

```

```

Networks Response {
  api_[1-9]+_v6 Network configs Response {
    subnet* string
    Subnet CIDR

    gateway string
    Gateway IP

    vlan_id integer
    VLAN ID
  }
}

```

```

    api_[1-9]
    +_v4* Network configs Response {
        subnet* string
            Subnet CIDR

        gateway string
            Gateway IP

        vlan_id integer
            VLAN ID
    }

    management_
    [1-9]+_v6 Network configs Response {
        subnet* string
            Subnet CIDR

        gateway string
            Gateway IP

        vlan_id integer
            VLAN ID
    }

    management_
    [1-9]+_v4* Network configs Response {
        subnet* string
            Subnet CIDR

        gateway string
            Gateway IP

        vlan_id integer
            VLAN ID
    }
}

Job Show Response {
    tasks* [Task Response {
        status* string
        updated_at* string($date-time)
        server* string
        command* string
        error* string
        started_at* string($date-time)
        stage* string
    }]

    description* string
    created_at* string($date-time)
    updated_at* string($date-time)
    aborted_at* string($date-time)
    version* string
    command* string
    log* string
    error* string
    started_at* string($date-time)
    stage* string
}

Cluster List Response {
    <cluster_name>* Cluster Info Response {
        info* string
            minLength: 1
            maxLength: 50
            Cluster info

        status* string
    }
}

Node List Response {
    <node_name>* Node Info Response {
        status* string
    }
}

Node Interface {
    api_[1-9]+_v6 string
        API IPv6 CIDR

    api_[1-9]
    +_v4* string
        API IPv4 CIDR

    management_
    [1-9]+_v6 string
        Management IPv6 CIDR

    management_
    [1-9]+_v4* string
        Management IPv4 CIDR
}

```

```

Baremetal Cluster Payload  {
  info*      string
             minLength: 1
             maxLength: 50
             Cluster info

  name*     string
             minLength: 1
             maxLength: 32
             Cluster name

  servers*  [Baremetal Node Payload  {
    name*    string
             minLength: 1
             maxLength: 32
             Node name

    oob_ip*  string
             OOB/CIMC IP

    oob_username  string
             OOB/CIMC username

    domain_name_servers  [string]
    oob_password  string
                     OOB/CIMC password

    flavor  string
           Flavor name

    boot_network  string
                 Boot Network interface

    ip_address*  Baremetal Node Interface Payload  {
      api_[1-9]  integer
                 API VLAN ID
      +_vlan_id

      api_[1-9]  string
                 API IPv4 Gateway
      +_gateway_v4*

      api_[1-9]+_v6  string
                     API IPv6 CIDR

      api_[1-9]  string
                 API IPv4 CIDR
      +_v4*

      api_[1-9]  string
                 API IPv6 Gateway
      +_gateway_v6

      management_[1-9]+_v6  string
                             Management IPv6

      management_[1-9]  string
                         Management IPv4 Gateway
      +_gateway_v4
      management_[1-9]+_v4*  string
                              Management IPv4

      management_[1-9]  string
                         Management IPv6 Gateway
      +_gateway_v6
    }

    password_hash  string
                   Password Hash
  }
}

```

```

Site  {
  info*      string
             minLength: 1
             maxLength: 50
             Site info

  clusters*  [Cluster  {
    info*    string
             minLength: 1
             maxLength: 50
             Cluster info

    name*    string
             minLength: 1
             maxLength: 32
             Cluster name

    servers*  [Node  {
      name*    string
             minLength: 1
             maxLength: 32
             Node name

      oob_ip*  string
             OOB/CIMC IP

      oob_username*  string
                     OOB/CIMC username

      domain_name_servers  [string]
      oob_password*  string
    }
  }
}

```

```

                                OOB/CIMC password
                                flavor*      string
                                                Flavor name
                                boot_network  string
                                                Boot Network interface
                                ip_address*   Node Interface {
                                                api_[1-9]+_v6 string
                                                API IPv6 CIDR
                                                api_[1-9]
+_v4*      string
                                                API IPv4 CIDR
                                                management_
[1-9]+_v6  string
                                                Management IPv6 CIDR
                                                management_
[1-9]+_v4* string
                                                Management IPv4 CIDR
                                                }
                                password_hash string
                                                Password Hash
                                }
                                }
networking* Networking {
  ntp_servers* [string]
                description: NTP IP
  domain_name_servers* [string]
                description: DNS IP
  domain_name* string
                Domain Name
  networks* Networks {
                api_[1-9]+_v6 Network Interface {
                subnet* string
                Subnet CIDR
                gateway string
                Gateway IP
                vlan_id integer
                VLAN ID
                }
                api_[1-9]
+_v4*      Network Interface {
                subnet* string
                Subnet CIDR
                gateway string
                Gateway IP
                vlan_id integer
                VLAN ID
                }
                management_
[1-9]+_v6  Network Interface {
                subnet* string
                Subnet CIDR
                gateway string
                Gateway IP
                vlan_id integer
                VLAN ID
                }
                management_
[1-9]+_v4* Network Interface {
                subnet* string
                Subnet CIDR
                gateway string
                Gateway IP
                vlan_id integer
                VLAN ID
                }
                }
  }
  name* string
        minLength: 1
        maxLength: 32
        Site name
  metadata* Metadata {
                ssh_key* string
                password_hash* string
                Password Hash
                }
}

Site List Response {
  <site_name>* Site Info Response {

```

```

        info*      string
                   minLength: 1
                   maxLength: 50
                   Site info
      status*     string
    }
  }

Cluster {
  info*      string
             minLength: 1
             maxLength: 50
             Cluster info

  name*     string
            minLength: 1
            maxLength: 32
            Cluster name

  servers*  [Node {
              name*      string
                       minLength: 1
                       maxLength: 32
                       Node name

              oob_ip*    string
                       OOB/CIMC IP

              oob_username* string
                       OOB/CIMC username

              domain_name_servers [string]
              oob_password*      string
                       OOB/CIMC password

              flavor*     string
                       Flavor name

              boot_network string
                       Boot Network interface

              ip_address*  Node Interface {
                           api_[1-9]+_v6 string
                               API IPv6 CIDR

                           api_[1-9]
                           +_v4* string
                               API IPv4 CIDR

                           management_
                           [1-9]+_v6 string
                               Management IPv6 CIDR

                           management_
                           [1-9]+_v4* string
                               Management IPv4 CIDR
                           }

              password_hash string
                       Password Hash
            }]]
}

Site Response {
  info*      string
             minLength: 1
             maxLength: 50
             Site info

  status*    string

  networking* Networking Response {
              ntp_servers* [string]
                       description: NTP IP

              domain_name_servers* [string]
                       description: DNS IP

              domain_name* string
                       Domain Name

              networks* Networks Response {
                       api_[1-9]+_v6 Network configs Response {
                           subnet* string
                               Subnet CIDR

                           gateway string
                               Gateway IP

                           vlan_id integer
                               VLAN ID
                       }

                       api_[1-9]
                       +_v4* Network configs Response {
                           subnet* string
                               Subnet CIDR

                           gateway string
                               Gateway IP
                       }
            }
}

```

```

        vlan_id      integer
                    VLAN ID
    }
    management_
[1-9]+_v6 Network configs Response {
        subnet*    string
                    Subnet CIDR

        gateway    string
                    Gateway IP

        vlan_id    integer
                    VLAN ID
    }
    management_
[1-9]+_v4* Network configs Response {
        subnet*    string
                    Subnet CIDR

        gateway    string
                    Gateway IP

        vlan_id    integer
                    VLAN ID
    }
}
}
name*        string
            minLength: 1
            maxLength: 32
            Site name
clusters*   Clusters Response {
    <cluster_name>* Cluster Config Response {
        info*      string
                    minLength: 1
                    maxLength: 50
                    Cluster info

        status*    string

        servers*   Nodes Response {
            <node_name>* Node Response {
                status*      string

                oob_ip*      string
                            OOB/CIMC IP

                oob_username* string
                            OOB/CIMC username

                domain_name_servers [string]

                oob_password* string
                            OOB/CIMC password

                flavor*       string
                            Flavor name

                boot_network  string
                            Boot Network interface

                ip_address*   Node Interface Response {
                    api_[1-9]+_v6 string
                                API IPv6 CIDR

                    api_[1-9]
+_v4* string
                                API IPv4 CIDR

                    management_
[1-9]+_v6 string
                                Management IPv6
                                CIDR

                    management_
[1-9]+_v4* string
                                Management IPv4
                                CIDR
                }

                password_hash string
                            Password Hash
            }
        }
    }
}
}
metadata*   Metadata Response {
    ssh_key*  string

    password_hash* string
                Password Hash
}
}

Running Jobs Response {
    <job_uuid>* Job Show Response {
        tasks* [Task Response {
            status*      string

            updated_at*  string($date-time)
        }
    }
}
}

```

```

        server*      string
        command*    string
        error*       string
        started_at* string($date-time)
        stage*       string
    }]
    description*    string
    created_at*     string($date-time)
    updated_at*     string($date-time)
    aborted_at*     string($date-time)
    version*        string
    command*        string
    log*            string
    error*          string
    started_at*     string($date-time)
    stage*          string
}
}

```

#### Baremetal Site Payload {

```

info*      string
minLength: 1
maxLength: 50
Site info

```

#### clusters\* [Baremetal Cluster Payload {

```

info*      string
minLength: 1
maxLength: 50
Cluster info

```

```

name*      string
minLength: 1
maxLength: 32
Cluster name

```

#### servers\* [Baremetal Node Payload {

```

name*      string
minLength: 1
maxLength: 32
Node name

```

```

oob_ip*    string
OOB/CIMC IP

```

```

oob_username string
OOB/CIMC username

```

```

domain_name_servers [string]

```

```

oob_password string
OOB/CIMC password

```

```

flavor      string
Flavor name

```

```

boot_network string
Boot Network interface

```

#### ip\_address\* Baremetal Node Interface Payload {

```

api_[1-9]      integer
+_vlan_id      API VLAN ID

```

```

api_[1-9]      string
+_gateway_v4*  API IPv4 Gateway

```

```

api_[1-9]+v6  string
API IPv6 CIDR

```

```

api_[1-9]      string
+_v4*          API IPv4 CIDR

```

```

api_[1-9]      string
+_gateway_v6   API IPv6 Gateway

```

```

management_    string
[1-9]+v6       Management IPv6

```

```

management_    string
[1-9]          Management IPv4 Gateway
+_gateway_v4   Management IPv4 Gateway

```

```

management_    string
[1-9]+v4*      Management IPv4

```

```

management_    string
[1-9]          Management IPv6 Gateway
+_gateway_v6   Management IPv6 Gateway

```

```

}
password_hash string
Password Hash

```

```

}}]

```

```

name*      string
minLength: 1
maxLength: 32
Site name

```

#### common\_info\* Baremetal Common Info Payload {

```

ntp_servers [string]
description: NTP IP(s)

```



```

        oob_username*      string
                           OOB/CIMC username

        domain_name_servers* [string]
                           description: DNS IP(s)

        ssh_key*           string
        domain_name*       string
                           Domain Name

        oob_password*      string
                           OOB/CIMC password

        flavor*            string
                           Flavor name

        password_hash*     string
                           Password Hash
    }

Networks {
    api_[1-9]+_v6 Network Interface {
        subnet*      string
                    Subnet CIDR

        gateway      string
                    Gateway IP

        vlan_id      integer
                    VLAN ID
    }

    api_[1-9]
+_v4* Network Interface {
        subnet*      string
                    Subnet CIDR

        gateway      string
                    Gateway IP

        vlan_id      integer
                    VLAN ID
    }

    management_
[1-9]+_v6 Network Interface {
        subnet*      string
                    Subnet CIDR

        gateway      string
                    Gateway IP

        vlan_id      integer
                    VLAN ID
    }

    management_
[1-9]+_v4* Network Interface {
        subnet*      string
                    Subnet CIDR

        gateway      string
                    Gateway IP

        vlan_id      integer
                    VLAN ID
    }
}

```